Using direction and elapsed-time information to reduce the wireless cost of locating mobile units in cellular networks

Yitzhak Birk [a] and Yaron Nachman [b,*]

[a] *Electrical Engineering Department, Technion – Israel Institute of Technology, Haifa 32000, Israel*
[b] *Lannet Data Communications Ltd., Tel-Aviv 61131, Israel*

Received February 1995; revised August 1995

Abstract. This paper proposes the use of easily-obtainable information to reduce the "wireless" cost of locating mobile units in cellular communication networks. This comprises the direct cost of searching for them in different cells upon arrival of calls, as well as that of occasional position-updates issued by the units to reduce the number of cells that need to be searched. The direction of motion at the time of last update is used to construct an asymmetric distance-based reporting boundary and, in conjunction with the elapsed time since the latest position-update, to optimize the search order. For a Markovian motion model along a straight line and known motion parameters, optimal algorithms are provided. The results suggest that substantial savings may be attained in common demanding situations such as commuter traffic in congested corridors.

1. Introduction

An important problem in cellular communication networks is locating a mobile unit to receive a call or message. This paper focuses on the "wireless" component of this cost, which includes transmissions for the purpose of determining whether the requested unit is in a given cell, as well as occasional transmissions by the mobile units, in which they report their position and thereby reduce the direct cost of searches. There is clearly a trade-off between the two components, which depends both on the relative cost of the two types of transmissions and on the usage pattern. In this paper, we do not attempt to assign weights to the two components. Rather, we attempt to reduce both.

Cellular communication networks have experienced explosive growth in the last several years. The initial dimension of growth was in geographic coverage, with the user population and level of use within any given area held down by high prices and/or a limited capacity for voice traffic. This phase, while impressive and challenging in many ways, did not present special difficulty to the control traffic over the wireless network, which was a small fraction of the "real" traffic.

Presently, reductions in both equipment prices and use fees are giving rise to an explosive growth of user populations and their level of use within any given geographic area. This trend, which is likely to continue, is forcing a reduction in cell sizes. For a given motion pattern of a mobile unit and a given rate of incoming calls per user, reduced cell sizes bring about an increase in the amount of *wireless* control traffic per cell. For example, if cell sizes are reduced so as to keep the number of mobile units per cell constant, and units update their position based on the distance from their last known location, the number of "find" messages per cell per unit time will increase linearly with the number of units per unit area. If, instead, cells are used as the measure of distance for the update policy, the update traffic per cell will increase. (The actual motion of the mobile units is not affected by the cell organization.)

With the proliferation of highly portable computers and communication appliances, and especially once digital communication technology is used in cellular networks, there is likely to be an explosion in data traffic, characterized by relatively frequent short sessions.

From the above, it is clear that the cost of wireless control traffic, i.e., traffic aimed at locating a mobile unit or reporting its position as opposed to carrying the actual communication or tracking location during a conversation, will become an important issue. The cost manifests itself in the form of bandwidth as well as energy, the latter applying primarily to the update messages. Indeed, the problem of reducing the "wireless" cost of tracking mobile units has recently received much attention [1–6].

In [1], the geographic area is partitioned into convex regions, and the cells on the boundary between any two such regions are designated as "reporting centers": a unit that enters such a cell must report its position. Using the taxonomy of [2], this scheme is "static" and "global" in that position updates by all units take place at the same fixed, predetermined subset of cells. The focus in [1] is on the optimization of the shape of the regions for minimal total update rate, given the motion parameters and the number of cells in a region.

The wireless resource is, to a first order, partitioned among the cells in an inflexible way. Consequently, the

* Work carried out at Technion.

"wireless" cost of control traffic should be evaluated on a per-cell basis, with the maximum (over cells) being perhaps a more representative measure than the aggregate traffic. The fact that, with global schemes, the update traffic is not divided uniformly among the cells, puts such schemes at an inherent disadvantage. Also, the fixed reporting cells prevent an optimal accommodation of mobile units with different motion parameters.

In [6], the authors explore a way of distributing the control load more uniformly among the cells. This is done by dividing the users into groups and assigning to each group a different grid of reporting cells. To avoid the problem of frequent updates by a user wandering near the boundary of one of his group's reporting cells, each group of users is actually assigned multiple grids of reporting cells, and a user switches to a different grid upon reporting. The two components of this approach are referred to as "multi-grouping" and "multi-switching", respectively.

In [5], the authors construct a registration area whose size is determined individually for each mobile unit at every position-update time. The size is determined according to the individual incoming-call- and update rates. For a registration area of $k \times k$ cells, a search that is carried out concurrently in all cells within this area at a cost proportional to the area, and assuming an update rate that is inversely proportional to k, the overall cost of updates and searches per user per unit time assumes the form:

$$C(k) \approx A \cdot k^2 + B\frac{1}{k},$$

where A is proportional to the rate of incoming calls per user and B is proportional to the update rate of a user with a $k \times k$ registration area. $C(k)$ is minimized by setting k equal to

$$k_{opt} \approx \sqrt[3]{\frac{B}{2A}}.$$

In [2], a comparison is carried out among update policies that use the position and time of the last update by any given unit as the reference point for that unit's next update. The elapsed time since the last update, the number of cell-changes, and the Euclidean distance from the last point of update are proposed as update criteria and compared. The conclusion is that distance-based update is generally superior to the others. Hybrid schemes are possible as well but are not considered. The studies assume Markovian motion on a ring topology. For a distance-based update policy, an explicit expression is provided for the update rate as a function of the update distance and the motion parameters. The trajectories of the mobile unit in different inter-update intervals are taken to be independent. The search for a mobile unit is assumed to be carried out sequentially, by decreasing order of steady-state probability of being in any given cell between the distance-based reporting boundaries. As is readily evident, the search thus begins from the cell of the latest update and progresses symmetrically in both directions. The cost of finding a unit is reported in terms of "search distance", so the number of single-cell searches is twice as high.

Given the relative cost of an update and a search in a single cell, the work in [3] incorporates the effect of frequent incoming calls and the resulting implicit position updates on the motion model and the optimal find–update tradeoff. However, the model can only handle static policies, the motion model is memoryless from step to step and hence also between updates, and searches proceed away from the latest known location, as in [2]. The work in [3] captures the notion of time in some sense by taking into account the effect of the frequency of incoming calls on the motion model, and by presenting an update criterion that is probabilistic in time. However, this criterion, as noted by the authors, is actually deterministic in distance. Thus, it is a distance-based update policy, with the optimal distance being a function of the motion parameters and the frequency of incoming calls. The search order is the same as in [2], so the cost of finding a unit at any given location is independent of the time elapsed since the last position update.

The current paper investigates the use of additional, easily obtainable information, to further customize the choice of the next set of reporting cells per mobile unit at the time of each position update and to choose the cell-order in which the unit is sought after in the event of an incoming call. The ideas are evaluated for the specific case of a Markovian motion model on a linear topology, which facilitates comparison with [2], but are useful in many other situations.

In section 2, we introduce a linear Markovian motion model that carries direction information over from one inter-update interval to the next, and derive an optimal asymmetric distance-based update criterion along with the resulting update frequency. In section 3, we derive for this model the optimal search strategy that exploits the knowledge of the elapsed time since the last update. Section 4 discusses the results, and section 5 offers concluding remarks.

2. An asymmetric distance-based position-update criterion

The occurrence of a position update, especially an explicit one, does not affect the motion of the mobile unit, but does reveal information about it. We next use knowledge of the direction of motion at the time of a position update to derive an optimal "asymmetric" distance-based update criterion for Markovian motion on a linear topology.

As in [2], we assume that the distance between the next possible reporting points, one to the left of the latest such point and one to its right, is $2D$ cells, where D is some constant. However, they needn't be symmetric about that point. Also, merely for facility of exposition, we elect to always denote the reporting cells by $-D$ and $+D$, and allow the starting point to vary within this range. Thus, the mobile unit begins its current journey in cell C, and reports its position once it enters cell $-D$ or $+D$. When a unit enters one of the reporting cells, that cell is immediately relabeled "C", a new pair of reporting cells is chosen, and a new period is started. Therefore, the unit can only be found in the range $[-D+1, D-1]$. Finally, without loss of generality, we assume that the unit always begins its journey with rightward being the "forward" direction. (Since its direction at the latest update time is known, this is only a matter of notation.) Our goal in this section is to pick C based on the motion parameters so as to minimize the update rate. For convenience, we use a slotted time model.

Until it enters one of the reporting cells, the unit's motion is governed by the following probabilities:

- q — probability that the next step will be in the *same* direction.
- v — probability that the next step will be in the *opposite* direction.
- $1-q-v$ — probability of *stopping*.
- p_1 — probability of *resuming* motion in the *same* direction as before stopping.
- p_2 — probability of *resuming* motion in the *opposite* direction.
- $1-p_1-p_2$ — probability of *not resuming* motion in the next time slot.

The mobile unit can be in one of four states, which are depicted in Fig. 1 along with the transition probabilities. The states are:

- R — the unit will move rightward one cell in the next time slot.
- L — the unit will move leftward one cell in the next time slot.
- SR — the unit stays put; its most recent move was rightward.
- SL — the unit stays put; its most recent move was leftward.

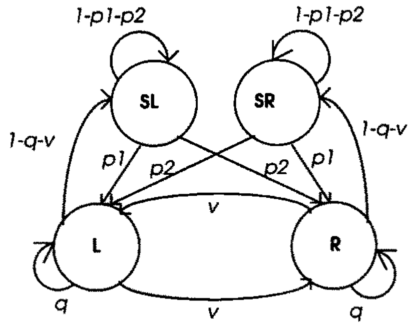

Fig. 1. Markovian-motion state machine.

Note that the definition of a state includes the action in the next time slot. Thus, being in state L or R at time n implies being one cell to the left or right of the current location, respectively, at time $n+1$. In contrast, being in state SL at time n and moving to state R implies remaining at the same location at time $n+1$ and being one cell to the right at time $n+2$. Similarly for other states and state-transitions.

An important difference between this model and that of [2] is the parameterization of the "persistence" of a unit across stops. (In [2], it was implicitly assumed that a unit loses its sense of direction once it stops, corresponding to $p_1 = p_2$.)

The Markov chain depicted in Fig. 2 describes the possible (location, state) combinations for a mobile unit, as well as the transition probabilities. With the exception of the ends of the reporting interval and the associated arcs, the chain has a repetitive structure. Arcs from $(-(D-1), L)$ to (C, R), (C, L) and (C, SR) all represent position updates at the left end of the interval, followed by the establishment of a new reporting boundary. Those from $(D-1, R)$ similarly correspond to updates at the right end of the interval. The choice of "rightward" as the forward direction (corresponding to no direction change) at position-update time manifests itself in identical arcs and arc weights from $(-(D-1), L)$ and from $(D-1, R)$ to $(C, ^*)$, and in the absence of arcs from these locations to (C, SL).

Let $Q_{d,s}$ denote the steady-state probability that a unit is at location d in state $s \in \{R, L, SR, SL\}$. The model can then be described by the following balance equations:

$$Q_{C,R} = q \cdot Q_{-(D-1),L} + q \cdot Q_{D-1,R} + q \cdot Q_{C-1,R} + v \cdot Q_{C+1,L} + p_1 \cdot Q_{C,SR} + p_2 \cdot Q_{C,SL}, \quad (1)$$

$$Q_{C,L} = v \cdot Q_{-(D-1),L} + v \cdot Q_{D-1,R} + v \cdot Q_{C-1,R} + q \cdot Q_{C+1,L} + p_2 \cdot Q_{C,SR} + p_1 \cdot Q_{C,SL}, \quad (2)$$

$$Q_{d,R} = q \cdot Q_{d-1,R} + v \cdot Q_{d+1,L} + p_1 \cdot Q_{d,SR} + p_2 \cdot Q_{d,SL}, \quad d \neq -(D-1), C, (D-1), \quad (3)$$

$$Q_{d,L} = v \cdot Q_{d-1,R} + q \cdot Q_{d+1,L} + p_1 \cdot Q_{d,SL} + p_2 \cdot Q_{d,SR}, \quad d \neq -(D-1), C, (D-1), \quad (4)$$

$$Q_{d,SR} = (1-q-v) \cdot Q_{d-1,R} + (1-p_1-p_2) \cdot Q_{d,SR}, \quad d \neq -(D-1), C, \quad (5)$$

$$Q_{d,SL} = (1-q-v) \cdot Q_{d+1,L} + (1-p_1-p_2) \cdot Q_{d,SL}, \quad d \neq (D-1), \quad (6)$$

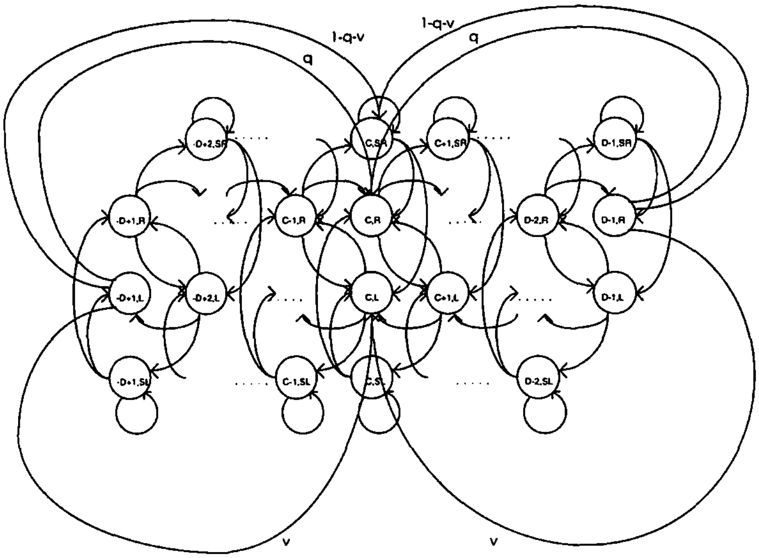

Fig. 2. Markov chain describing the motion model.

$$Q_{C,SR} = (1-q-v)\cdot Q_{D-1,R} + (1-q-v)\cdot Q_{-(D-1),L} + (1-q-v)\cdot Q_{C-1,R} + (1-p_1-p_2)\cdot Q_{C,SR}\,, \tag{7}$$

$$Q_{D-1,R} = q\cdot Q_{D-2,R} + p_1\cdot Q_{D-1,SR}\,, \tag{8}$$

$$Q_{-(D-1),R} = v\cdot Q_{-(D-2),L} + p_2\cdot Q_{-(D-1),SL}\,, \tag{9}$$

$$Q_{D-1,L} = v\cdot Q_{D-2,R} + p_2\cdot Q_{D-1,SR}\,, \tag{10}$$

$$Q_{-(D-1),L} = q\cdot Q_{-(D-2),L} + p_1\cdot Q_{-(D-1),SL}\,, \tag{11}$$

$$\sum_{d=-(D-1)}^{D-1} (Q_{d,R} + Q_{d,L} + Q_{d,SR} + Q_{d,SL}) = 1\,, \tag{12}$$

$$\text{UPDATE} = Q_{-(D-1),L} + Q_{D-1,R}\,. \tag{13}$$

The update rate, derived in Appendix A, is,

$$\text{UPDATE} = \frac{\hat{q}}{(D-C)[(D+C)(1-\hat{q}) + 2\hat{q} - 1]}\,\frac{1}{T}\,, \quad -(D-1) < C < (D-1)\,, \tag{14}$$

where

$$\hat{q} = q + (1-q-v)\frac{p_1}{p_1+p_2}\,, \quad 0 < \hat{q} \leqslant 1\,, \tag{15}$$

and

$$T = 1 + (1-q-v)\frac{1}{p_1+p_2} = 1\cdot q + 1\cdot v + \left(1 + \frac{1}{p_1+p_2}\right)(1-q-v)\,. \tag{16}$$

Note that T represents the mean holding time in a cell, and the mean update rate can thus be expressed as the product of this time and the number of cell changes between updates.

The foregoing set of equations is only valid in the range $-(D-1) < C < (D-1)$. The case of $C = D-1$ makes no practical sense, but that of $C = -(D-1)$ is important, since this is the likely optimal starting point when motion persistence is sufficiently high. The Markov chain for this special case differs from that of Fig. 2 in that the arcs from the left edge to C originate and end at the same cell. The update rate, whose derivation is omitted for brevity but can be found in [7], is

$$\text{UPDATE} = \frac{1}{(2D-1)}\,\frac{1}{T}\,, \quad C = -(D-1)\,, \quad q > 0\,. \tag{17}$$

Note. When $C = -D+1$ and $q > 0$, the mean number of cell traversals between position updates is independent of the motion parameters. This is quite surprising, and is not true of higher moments of the number of traversals.

Finally, the value of C that minimizes the update rate is

$$C_{opt} = \max\left\{1 - \frac{1}{2(1-\hat{q})}\,, -(D-1)\right\}.$$

It is interesting to observe that this value of C_{opt} closely corresponds to placing the starting point to the "left" of the center of the reporting zone by a number of cells equal to one half of the expected number of cells that the unit traverses before reversing its direction. This result is quite intuitive, since after so doing the mobile unit sees the reporting boundaries at the same positions (relative to its own position and direction) as immediately following a position update. Finally, C must be within the registration area range $[-D+1, D-1]$, so there is a maximum value of q beyond which $C_{opt} = -D+1$.

Fig. 3 depicts the update rate as a function of the directional persistence while in motion, $q/(q+v)$. It can readily be seen that the benefit from optimizing the positioning of the reporting boundaries increases with an increase in directional persistence, and the maximum advantage is by a factor of two. Plots are shown for persistent resumption (a) and for random resumption (b), for a single value of $(q+v)$. Plots for other values are similar [7]. Finally, we note that as cells become smaller while the actual motion patterns do not change, the case of highly persistent motion (in terms of cell changes) will become very common.

The curves of Fig. 3 also suggest that one can closely approximate the optimal choice of C with a simpler one, whereby $C = 0$ if persistence is below a certain value and $C = -(D-1)$ otherwise. The following two lemmas determine the crossover point and quantify the closeness of the approximation.

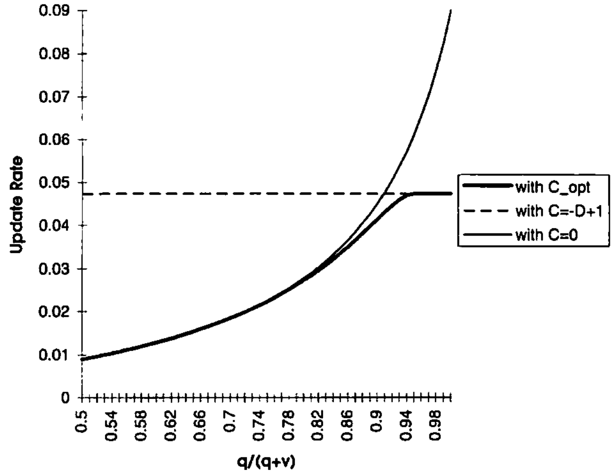

Fig. 3(a). Update rate vs. $q/(q+v)$; $q+v=0.9$; persistent resumption; $D=10$.

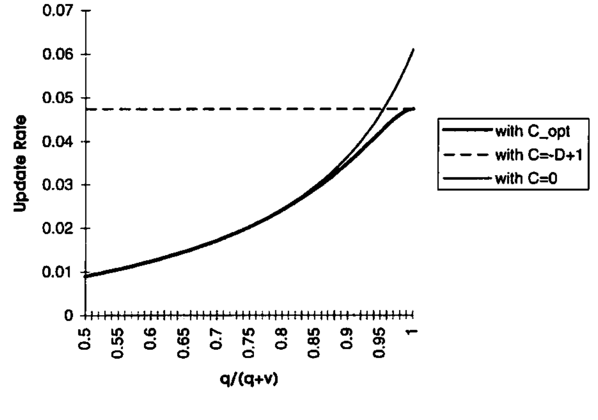

Fig. 3(b). Update rate vs. $q/(q+v)$; $q+v=0.9$; non-persistent (random) resumption; $D=10$.

Lemma. The bi-valued approximation of C_{opt} is closest if $C=0$ whenever

$$q + (1-q-v)\frac{p_1}{p_1+p_2} < \frac{D}{D+1},$$

and $C = (-D+1)$ otherwise.

Proof. Equating the update rate for $C=0$ (14) with that for $C=(-D+1)$ (17) yields the crossover point:

$$\hat{q} = q + (1-q-v)\frac{p_1}{p_1+p_2} = \frac{D}{D+1}. \qquad \square$$

Lemma. The maximum relative error in the bi-valued approximation is

$$\Delta = \frac{(D-1)^2}{4D(2D-1)} < 0.125.$$

Proof. The relative error is given by

$$\Delta = \left(\frac{\text{UPDATE}(\hat{q}, C=0)}{\text{UPDATE}(\hat{q}, C=C_{opt})}\right) - 1.$$

It can readily be observed from Fig. 3 (and proven formally) that the maximum error occurs at the crossover point. At this point,

$$C_{opt} = 1 - \frac{1}{2(1-\hat{q})} = -\frac{D-1}{2}.$$

Substituting in (14) completes the proof. $\square$

3. Using elapsed-time information to optimize the search order

For each incoming call, the system initiates a search for the requested mobile unit within that unit's current registration area. This process may be carried out in parallel in multiple cells, or sequentially in one cell at a time. The mean number of cells in which the unit is sought has been referred to as the "find" cost. The discussion in this paper is restricted to sequential search.

The probability distribution of the location of a mobile unit within the area bounded by the next reporting cells changes with the passage of time since an update. As depicted in Fig. 4, it is not even true that the most likely location at all times is the latest reporting cell and that likelihood decreases as one moves away from this cell. Accordingly, the optimal search strategy is not stationary, and the minimum cost of finding a unit is a function of elapsed time even for any given current location. We next develop optimal search schemes which, unlike the stationary ones of [2] and [3], exploit the elapsed-time information, and show that they lead to a substantial reduction in the find cost.

To derive the mean find cost, we first derive it for fixed values of the discrete elapsed time, denoted by n. We

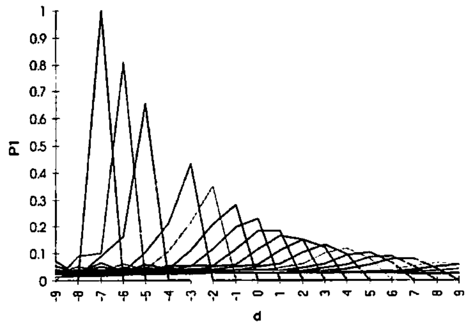

Fig. 4. Probability distribution of a mobile unit's location at different elapsed times since its latest position update. $q/(q+v)=0.9$; $q+v=0.9$; $D=10$. The initial direction of the unit is rightward. The curves shift rightward with an increase in elapsed time, and become "shallower".

then compute the probability that at the time of a randomly-arriving incoming call, the elapsed time is n. Finally, we compute the inner product of these two vectors. The effect of implicit position updates that result from successful searches is ignored. Following is a more detailed outline of the derivations.

Referring to the 4-state diagram in Fig. 1, let $\hat{Q}_{d,n,s}$ denote the probability that at elapsed time n since the latest position update, the unit is located at d and is in state $s \in \{L, R, SL, SR\}$. $\hat{Q}_{d,n,s}$ is, of course, also a function of C, but this is omitted for simplicity of notation.

By definition, $\hat{Q}_{C,0,R}=q$, $\hat{Q}_{C,0,L}=v$, $\hat{Q}_{C,0,RS}=1-q-v$, $\hat{Q}_{C,0,SL}=0$, and $\hat{Q}_{d,0,*}=0$ for $d \neq C$. Also, a rightward step from $d=D-1$ and a leftward one from $d=-(D-1)$ cause a position update and reset the unit to $(d=C, n=0)$. $\hat{Q}_{d,n,s}$ can be computed using the known initial value (at $n=0$) and the state-transition probabilities in Fig. 2.

Next, let, $P_1(d,C,n)$, $d \in [-(D-1),(D-1)]$, denote the probability that the unit is at location d at time n given that it was located at $d=C$ at time 0. Clearly, $P_1(d,C,n)$ is simply the sum of $\hat{Q}_{d,n,s}$ over the four states. The computation of the optimal sequential search order and of the mean number of single-cell searches required in order to locate the unit (the "find cost") proceeds according to the following algorithm:

Algorithm 1: find cost with the optimal time-dependent search order

1. Given C, D, q, v, p_1 and p_2, construct $\hat{Q}_{d,n,s}$ by "rolling" the system forward in discrete time steps using the 4-state machine and transition probabilities, from $n=0$ until such n that the probability of not yet having updated position is sufficiently small, say $\epsilon=0.0001$. We denote this value of n by n_0. At each step, also compute $P_1(d,C,n)$. (n_0 can also be obtained in a similar manner.)
   Note. Since $\hat{Q}_{d,n,s}$ and $P_1(d,C,n)$ are conditioned upon the elapsed time being n, i.e., upon no update and resetting of the state having occurred prior to n, a simple normalization is required at each time step.

2. For each $n \in 0,1,\ldots,n_0$, sort the $2D-1$ values of $P_1(d,C,n)$ corresponding to $d=-(D-1)$, ..., $D-1$ in descending order, and assign them (in order) to $P_2(x,C,n)$, where $x=1$, ..., $2D-1$. ($P_2(x,C,n)$ represents the optimal search order for this unit at time n.)

3. For each n, calculate the mean number of searches:
$$\begin{cases} \bar{N}_{opt}(n,C)=\sum_{x=1}^{2D-1} x \cdot P_2(x,C,n)\,, & n=0,1,\ldots n_0\,, \\ \bar{N}_{opt}(n,C) \leqslant D\,, & n>n_0\,. \end{cases}$$

4. For $n=0\ldots n_0$, calculate the probability of a call arriving for a unit at elapsed time n:
$$P_{call\_arrival}(n,C) = \begin{cases} Update(C)\,, & n=0\,, \\ P_{call\_arrival}(n-1,C)-\hat{Q}_{-(D-1),n-1,L}-\hat{Q}_{D-1,n-1,R}\,, & n\geqslant 1\,. \end{cases}$$

5. Finally, the minimum (over all time-dependent search orders) find cost is given by
$$\bar{N}_{opt}(C) \cong \sum_{n=0}^{n_0} P_{call\_arrival}(n,C)\cdot \bar{N}(n,C)+\epsilon \cdot D\,,$$
   where the error is at most $\epsilon \cdot D$ and can easily be made negligible.

Fig. 5 depicts the mean number of single-cell searches as a function of $q/(q+v)$ for various choices of C. The two figures, (a) and (b), differ in both the persistence of motion and that of resumption. In each case, the curves in the top set correspond to search orders that are based on the aggregate probability distribution of the unit's location, whereas those in the bottom set correspond to the search orders that are optimized based on the elapsed time from the latest position update. The significant advantage of the time-based search orders is self evident. However, the advantage at points of low directional persistence (e.g., $q=v$) and rare stops ($q+v$ close to 1) is largely due to an artifact of the discrete nature of the model, namely the fact that the unit can either be only at odd values of d or only at even ones, depending on n and C. The advantage increases (and is real) as the directional persistence increases. Another important observation is that, while in the case of fixed search orders there is a conflict between the optimization of C for update rate and for find cost, this conflict nearly disappears with the use of time-based search orders, permitting us to optimize C for minimum update rate. Behavior with other parameter values is similar [7]. To fully understand the behavior of the different curves, let us next review the exact models underlying them. We refer to Fig. 5(a), but the explanations also apply to (b).

In all curves, the motion model carries direction

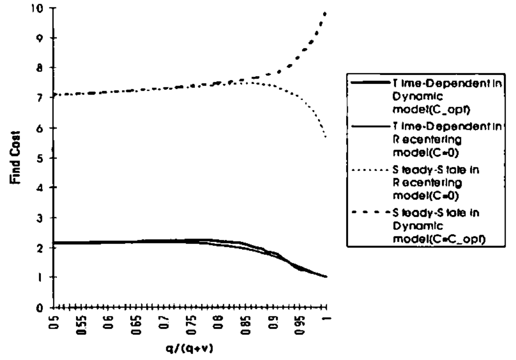

Fig. 5(a). Mean no. of single-cell searches vs. $q/(q+v)$; $q+v=1.0$; persistent resumption; $D=10$.

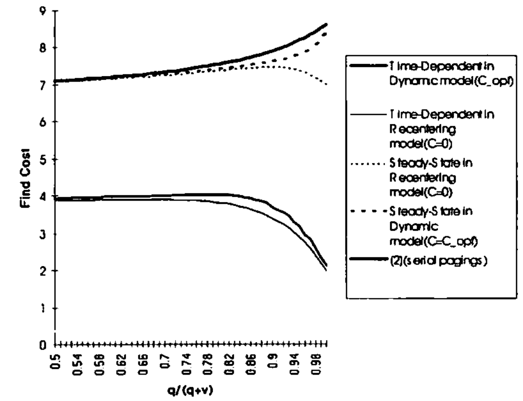

Fig. 5(b). Mean no. of single-cell searches vs. $q/(q+v)$; $q+v=0.9$; $D=10$; non-persistent resumption.

across updates, and the direction of motion at the time of update is known. In the steady-state curves, the search order does not depend on the elapsed time, but the direction is nevertheless used to choose between two search orders which are "mirror images" of each other. In the time-dependent cases, the elapsed time when the call arrives is used to further refine the search order by using the correct location-probability distribution for that time.

Consider initially the static scheme with $C=0$, assuming that the unit was headed rightward at the time of the last update. As $q/(q+v)$ increases, the probability that the unit is to the left of C diminishes, and the probability of location becomes uniform across the range $d>C$. The former phenomenon makes the search more efficient, while the latter one makes it less efficient. (Note that these are changes in steady state probabilities, not temporal changes!) The fact that the cost comes down as $q/(q+v)$ approaches one suggests that the former phenomenon prevails. When $q=1$, all positions to the right of C are equiprobable, and the mean number of searches is $D/2$.

When C is allowed to vary so as to minimize the update rate, it moves leftward as $q/(q+v)$ increases, eventually equaling $-(D-1)$. As $q/(q+v)$ increases beyond this point, the "bad" foregoing phenomenon still takes place, but the "good" one does not, since there are no remaining locations to the left of C. Consequently, the cost rises, eventually reaching $(2D-1)/2$, because the entire range of locations is equiprobable.

The model in [2] does not exploit the knowledge of the direction at the time of the latest position update, and holds $C=0$. The location-probability distribution can be thought of as the sum of the "directional" one and its reflection, divided by two for normalization. A ramification of this is that only the "bad" phenomenon (uniformization of the probability distribution) takes place, leading to the highest cost across the range, with equality to the case of update-optimized C when $q/(q+v)=1$. This is depicted as an additional curve in Fig. 5(b).

With the optimal time-dependent search orders, the "good" phenomenon still occurs as $q/(q+v)$ increases. The "bad" one, namely the uniformization of the probability distribution to the right of C is only applicable to the steady state location probability distribution. Given the elapsed time, the distribution actually becomes narrower around the expected location, and the mean number of searches thus decreases. As $q/(q+v)$ approaches one, the journey is nearly deterministic and the mean number of single-cell searches approaches one. This represents a reduction in find cost by a factor of up to $D/2$ relative to the best time-independent search order, and by up to a factor of D relative to a search order that does not make use of the direction at the most recent update.

4. Discussion

The use of direction information at position-updates to construct optimal asymmetric distance-based reporting boundaries was shown to reduce the update rate by up to a factor of two. The reduction becomes more pronounced as the directional persistence increases. The numerical results indicate that for low and moderate levels of persistence in the direction of motion, it suffices to position the new reporting boundaries symmetrically about the latest known location. Beyond a certain point, one may as well go directly to the other extreme and position one boundary adjacent to the last known location while putting the other one as far as possible from it in the initial direction of motion.

Optimization of the search order based on the elapsed time since the last position update was shown to yield a very substantial reduction in the mean required number of searches, as compared with a search strategy based on steady-state location probability distributions. The reduction increases with an increase in the directional persistence of the motion, reaching a factor of $D/2$.

With fixed search orders and a given distance between the next reporting cells, choosing those cells so as to minimize the update rate conflicted with minimization of the cost of finding a unit. However, when using the search strategies that take into account the elapsed time since the last update, the required number of searches is far less sensitive to the placement of the reporting cells, which can be optimized for minimum update rate.

Throughout the paper, we assumed that the size of the registration area is given ($2D$), and explored ways of minimizing the amount of control traffic. One could similarly vary D while also optimizing the choice of C so as to attain a desired update rate while minimizing the size of the reporting zone and hence the required number of searches.

As the frequency of incoming calls increases, their influence on the motion model (through implicit updates) can no longer be neglected [3]. While we have been able to extend our model to include this influence and obtain results, the discussion here will be qualitative. As the probability of an implicit position update at any location increases, the rate of explicit updates decreases, as does the difference between the update rate with an optimal choice of C and with $C = 0$. The number of cells that need to be searched also decreases since long elapsed times, which lead to longer searches, are less likely. As for the optimal search order for any given elapsed time, the rate of incoming calls has no effect. The reason is that the mere fact that the elapsed time is n implies that there was no incoming call for the unit since the last (implicit or explicit) position update, so the probability distribution of its location at this time is the same as with no incoming calls.

The quantitative results presented in this paper should be taken with a grain of salt, since actual motion is not necessarily Markovian, and its parameters are not always known. As for the use of a linear topology, this is actually representative of important practical situations such as congested traffic corridors.

The additional information required for the optimizations can be obtained as follows. The elapsed time since the latest update can easily be recorded in the same database that records a unit's last known location. The direction of motion at the time of a position update, as well as more detailed information such as speed (whose knowledge is required for the effective use of the elapsed-time information), can be collected by the mobile unit between updates. It can then be included in the update message at negligible additional cost. In many important and challenging scenarios for a cellular network, such as a congested highway, pedestrian pathway or train, the speeds of most units are often similar if not identical, and very reliable information can be obtained by the system either from position updates or through other means such as sensors along the road. Finally, degradation of performance (i.e. increase in control traffic) with that of the estimates is graceful, so the fact that one cannot do the ultimate does not imply that nothing should be done.

5. Conclusions

The amount of wireless control traffic per cell in cellular communication networks is expected to increase as cells become smaller and short connection times, as for data traffic, become more common. This paper explored the usefulness of direction information and knowledge of the elapsed time since the last position update for reducing the "wireless" cost of tracking idle units, and showed that these can substantially reduce the amount of control traffic in the wireless network. While quantitative analysis was provided only for a Markovian model over a linear topology, the results are indicative of the potential benefits. Accordingly, the operators of cellular communication networks, especially in dense areas and challenging situations, should seriously consider the collection of information pertaining to the motion pattern of the units. This will be used in dynamically specifying the next reporting region for each unit and in deciding the order in which cells should be searched when a unit is to be located. (The actual algorithms may vary.) This investment in computing and other "wired" resources may significantly reduce the consumption of precious wireless resources for control traffic.

Appendix A

The size of the diagram in Fig. 2 varies with D, as does the cardinality of set of equations that describe it. To arrive at a set of equations with fixed cardinality, we proceed as follows:

Manipulation of (5) and (6) produces

$$Q_{d,SR} = \frac{1-q-v}{p_1+p_2} Q_{d-1,R} \tag{18}$$

and

$$Q_{d,SL} = \frac{1-q-v}{p_1+p_2} Q_{d+1,L} \,. \tag{19}$$

From (3) and (4), it follows that

$$Q_{d,R} = \hat{q} \cdot Q_{d-1,R} + (1-\hat{q}) \cdot Q_{d+1,L} \tag{20}$$

and

$$Q_{d,L} = (1-\hat{q}) \cdot Q_{d-1,R} + \hat{q} \cdot Q_{d+1,L} \,, \tag{21}$$

where

$$\hat{q} = q + (1-q-v)\frac{p_1}{p_1+p_2} \,. \tag{22}$$

Using boundary conditions, the recursive equations (20) and (21) can be replaced with the following four equations, which hold for $0 \leqslant \hat{q} \leqslant 1$:

$$Q_{d,L} = (D+d-1) \cdot Q_{-(D-2),L} - (D+d-2) \cdot Q_{-(D-1),L}$$

$$\{-(D-1) \leqslant d \leqslant C\} \,,$$

$$Q_{d,R} = (D+d-1)\cdot Q_{-(D-2),R} - (D+d-2)\cdot Q_{-(D-1),R}\,,$$
$$\{-(D-1)\leqslant d\leqslant C-1\}\,,$$
$$Q_{d,L} = (D-d-1)\cdot Q_{D-2,L} - (D-d-2)\cdot Q_{D-1,L}\,,$$
$$\{C+1\leqslant d\leqslant D-1\}\,,$$
$$Q_{d,R} = (D-d-1)\cdot Q_{D-2,R} - (D-d-2)\cdot Q_{D-1,R}\,,$$
$$\{C\leqslant d\leqslant D-1\}\,.$$

The above equations, along with the original ones, can now be used to write the complete, fixed-cardinality set of equations in $Q_{-(D-1),L}$, $Q_{-(D-2),L}$, $Q_{C-1,R}$, $Q_{C,R}$, $Q_{C,L}$, $Q_{C+1,L}$, $Q_{-(D-1),R}$, $Q_{-(D-2),R}$, $Q_{(D-1),R}$, $Q_{D-2,R}$, $Q_{D-1,L}$, $Q_{D-2,L}$.
The set of equations is:

$$Q_{C,R} = \hat{q}\cdot Q_{-(D-1),L} + \hat{q}\cdot Q_{D-1,R} + \hat{q}\cdot Q_{C-1,R} + (1-\hat{q})\cdot Q_{C+1,L}\,,$$

$$Q_{C,L} = (1-\hat{q})\cdot Q_{-(D-1),L} + (1-\hat{q})\cdot Q_{D-1,R} + \hat{q}\cdot Q_{C+1,L} + (1-\hat{q})\cdot Q_{C-1,R}\,,$$

$$Q_{C-1,R} = (1-\hat{q})\cdot Q_{C,L} + \hat{q}\cdot(D+C-3)\cdot Q_{-(D-2),R} - \hat{q}\cdot(D+C-4)\cdot Q_{-(D-1),R}\,,$$

$$Q_{C+1,L} = (1-\hat{q})\cdot Q_{C,R} + \hat{q}\cdot(D-C-3)\cdot Q_{D-2,L} - \hat{q}\cdot(D-C-4)\cdot Q_{D-1,L}\,,$$

$$Q_{C,L} = (D+C-1)\cdot Q_{-(D-2),L} - (D+C-2)\cdot Q_{-(D-1),L}\,,$$

$$Q_{D-1,R} = \hat{q}\cdot Q_{D-2,R}\,,$$

$$Q_{-(D-1),L} = \hat{q}\cdot Q_{-(D-2),L}\,,$$

$$Q_{D-1,L} = (1-\hat{q})\cdot Q_{D-2,R}\,,$$

$$Q_{-(D-1),R} = (1-\hat{q})\cdot Q_{-(D-2),L}\,,$$

$$Q_{D-2,L} = \hat{q}\cdot Q_{D-1,L} + 2(1-\hat{q})\cdot Q_{D-2,R} - (1-\hat{q})\cdot Q_{D-1,R}\,,$$

$$Q_{-(D-2),R} = \hat{q}\cdot Q_{-(D-1),R} + 2(1-\hat{q})\cdot Q_{-(D-2),L} - (1-\hat{q})\cdot Q_{-(D-1),L}\,,$$

$$\begin{aligned}\sum_{d=-(D-1)}^{D-1} &(Q_{d,R} + Q_{d,L} + Q_{d,SR} + Q_{d,SL})\\
&= \{(D+C-1)(D+C)\cdot Q_{-(D-2),L}\\
&\quad -(D+C-3)(D+C)\cdot Q_{-(D-1),L}\\
&\quad +(D-C-2)(D-C-1)\cdot Q_{D-2,L}\\
&\quad -(D-C-4)(D-C-1)\cdot Q_{D-1,L}\\
&\quad +(D+C-2)(D+C-1)\cdot Q_{-(D-2),R}\\
&\quad -(D+C-4)(D+C-1)\cdot Q_{-(D-1),R}\\
&\quad +(D-C-1)(D-C)\cdot Q_{D-2,R}\\
&\quad -(D-C-3)(D-C)\cdot Q_{D-1,R}\}\frac{T}{2} = 1\,,\end{aligned}$$

$$\mathrm{UPDATE} = Q_{-(D-1),L} + Q_{D-1,R}\,.$$

Solving the equations using Mathematica, we obtain

$$\mathrm{UPDATE} = \frac{\hat{q}}{(D-C)[(D+C)(1-\hat{q})+2\hat{q}-1]}\,\frac{1}{T}\,,$$

where

$$T = 1 + (1-q-v)\frac{1}{p_1+p_2}$$

and

$$\hat{q} = q + (1-q-v)\frac{p_1}{p_1+p_2}\,.$$

References

[1] A. Bar-Noy and I. Kessler, Tracking mobile users in wireless communications networks, *Proc. Infocom '93*, Paper 10b.4, San Francisco, CA (1993) pp. 1232–1239.

[2] A. Bar-Noy, I. Kessler and M. Sidi, Mobile users: To update or not to update?, *Proc. Infocom '94*, Toronto, Canada (1994) pp. 570–576.

[3] U. Madhow, M.L. Honig and K. Steiglitz, Optimization of wireless resources for personal communications mobility tracking, *Proc. Infocom '94*, Toronto, Canada (1994) pp. 577–584.

[4] S.T.S. Chia, Location registration and paging in a third generation mobile system, BT Technology 9 (4) (1991) 61–68.

[5] H. Xie, S. Tabbane and D.J. Goodman, Dynamic location area management and performance analysis, *Proc. VTC '93*, Secaucus, NJ (1993) pp. 536–539.

[6] S. Okasaka, S. Onoe, S.Yasuda and A. Maebara, A new location updating method for digital cellular systems, *Proc. 41st VTC Conference*, St. Louis, MO (1991) pp. 345–350.

[7] Y. Nachman, Reducing the wireless resource needed for tracking mobile users in cellular communication networks, M.Sc. Dissertation, Electrical Engr. Dept., Technion – Israel Inst. of Tech. (1994) [Hebrew + English abstract].

Yitzhak Birk received the B.Sc. (cum laude) and M.Sc. degrees from the Technion – Israel Institute of Technology in 1975 and 1981, respectively, and a Ph.D. degree from Stanford University in 1987, all in electrical engineering. From 1976 to 1981, he was project engineer in the Israel Defense Forces. From 1986 to 1991, he was with IBM at the Almaden Research Center, where he worked on parallel architectures, computer subsystems and passive fiber-optic interconnection networks. He is presently on the faculty of the Electrical Engineering Department at the Technion, and heads its Parallel Systems Laboratory. His current research interests include computer systems and subsystems, as well as communication networks, with a special emphasis on parallel systems. Much of his activity is presently focused on parallel storage and communication architectures for video servers. Dr. Birk is a member of the Institute of Electrical and Electronics Engineers.
E-mail: birk@ee.technion.ac.il
Web page: http://www-ee.technion.ac.il

Yaron Nachman was born in Haifa, Israel, in 1966. He received the B.Sc. (cum laude) and M.Sc. degrees in electrical engineering from the Technion – Israel Institute of Technology, Haifa, Israel, in 1989 and 1995, respectively. From 1989 to 1994 he worked as Communication Engineer in the IDF. He is presently working as a Computer Networks System Engineer at Lannet Data Communication Ltd. He is working in research and development of switching hubs for local area networks based on protocols such as Ethernet, Token-Ring, FDDI, and Wireless LAN. His current interests are in architecture and design of cellular networks, high speed networks and wireless LAN.
E-mail: yaronn@lannet.com